

一、下载安装FastDFS

1. 下载解压

```
1. sudo apt-get install -y g++ gcc unzip
2.
3. cd /usr/local/src
4. sudo wget https://github.com/happyfish100/fastdfs/archive/master.zip
5. sudo wget https://github.com/happyfish100/libfastcommon/archive/master.zip
6.
7. sudo unzip master.zip
8. sudo unzip master.zip.1 # 下载下来都是叫master.zip所以会重名成.1
```

2. 安装libfastdfs和fastdfs(有先后顺序)

```
1. # 安装libfastcommon-master
2. cd /usr/local/src/libfastcommon-master/
3. sudo ./make.sh
4. sudo ./make.sh install
5.
6. # 安装fastdfs
7. cd /usr/local/src/fastdfs-master/
8. sudo ./make.sh
9. sudo ./make.sh install
10.
11. # Tip: ./make.sh 出错则 sudo su - root 切换到root用户再执行即可
```

二. 配置

1. 复制配置文件

默认配置在/etc/fdfs/下，复制模版配置文件至/etc/fdfs
sudo cp /usr/local/src/fastdfs-master/conf/* /etc/fdfs/

2. 配置tracker

```
sudo mkdir -pv /data/fastdfs/tracker
sudo vim /etc/fdfs/tracker.conf # 修改下面配置
```

```
1. disabled=false # 启用配置文件
2. port=22122 # 设置 tracker 的端口号
3. base_path=/data/fastdfs/tracker # 设置 tracker 的数据文件和日志目录
4. http.server_port=8080 # 设置 http 端口号（内置的，不用管）
```

3. 运行tracker

```
1. # 启动/停止/重启 tracker
2. sudo fdfs_trackerd /etc/fdfs/tracker.conf start/stop/restart
3.
4. # 查看tracker是否运行
5. sudo netstat -lnp |grep 'fdfs_trackerd'
6. sudo lsof -i:22122
7.
8. # 查看日志确定启动成功
9. tail /data/fastdfs/tracker/logs/trackerd.log
10. [2016-07-27 14:56:15] INFO - FastDFS v5.08, base_path=/data/fastdfs/tracker,
11. run_by_group=, run_by_user=, connect_ \
12. timeout=30s, network_timeout=60s, port=22122, bind_addr=,
13. max_connections=256, accept_threads=1, work_threads=4, s \
14. tore_lookup=2, store_group=, store_server=0, store_path=0, reserved_storage_
15. space=10.00%, download_server=0, allow \
16. _ip_count=-1, sync_log_buff_interval=10s, check_active_interval=120s,
17. thread_stack_size=64 KB, storage_ip_changed_ \
18. auto_adjust=1, storage_sync_file_max_delay=86400s,
19. storage_sync_file_max_time=300s, use_trunk_file=0, slot_min_siz \
20. e=256, slot_max_size=16 MB, trunk_file_size=64 MB, trunk_create_file_advance
21. =0, trunk_create_file_time_base=02:00, \
22. trunk_create_file_interval=86400, trunk_create_file_space_threshold=20 GB,
23. trunk_init_check_occupying=0, trunk_in \
24. it_reload_from_binlog=0, trunk_compress_binlog_min_interval=0,
25. use_storage_id=0, id_type_in_filename=ip, storage_i \
26. d_count=0, rotate_error_log=0, error_log_rotate_time=00:00,
27. rotate_error_log_size=0, log_file_keep_days=0, store_s \
28. lave_file_use_link=0, use_connection_pool=0,
29. g_connection_pool_max_idle_time=3600s
```

4. 配置storage

```
sudo mkdir -pv /data/fastdfs/storage  
sudo vim /etc/fdfs/storage.conf
```

```
1. disabled=false # 启用配置文件  
2. group_name=group1 # 组名, 根据实际情况修改  
3. port=23000 # 设置 storage 的端口号  
4. base_path=/data/fastdfs/storage # 设置 storage 的日志目录 (需预先创建)  
5. store_path_count=1 # 存储路径个数, 需要和 store_path 个数匹配  
6. store_path0=/data/fastdfs/storage # 存储路径  
7. tracker_server=192.168.137.210:22122 # tracker 服务器的 IP 地址和端口号 (不能设置为127.0.0.1)  
8. http.server_port=80 # 设置storage上启动的http服务的端口号  
9. # 如果发现启动不了记得查看日志
```

5. 运行storage

```
sudo fdfs_storaged /etc/fdfs/storage.conf start/stop/restart  
查看23000端口, 并检查日志
```

6. fdfs_monitor 查看storage状态

使用 fdfs_monitor 来查看一下storage的状态: 状态为ACTIVE,说明已经成功注册到了tracker

```
sudo fdfs_monitor /etc/fdfs/storage.conf
```

三、测试上传、下载、删除、查看

1. 配置 /etc/fdfs/client.conf

修改配置文件: sudo vim /etc/fdfs/client.conf

```
1. base_path=/tmp # the base path to store log files
2. tracker_server=192.168.137.126:22122
3. http.tracker_server_port=80
```

2.上传/测试上传

上传文件：fdfs_upload_file [storage_ip:port] [store_path_index]
sudo fdfs_upload_file /etc/fdfs/client.conf ~/.bashrc

测试上传：fdfs_test 配置文件 upload 上传的文件
sudo fdfs_test /etc/fdfs/client.conf upload /etc/fdfs/anti-steal.jpg

3.下载

下载文件：fdfs_download_file [local_filename] []
例如：fdfs_download_file /etc/fdfs/client.conf
group1/M00/00/00/wKgBCFeYKNGAG6DtAACc5I4i_bE824.zip newname.zip

4.查看

格式：fdfs_file_info 配置文件 存储的路径
sudo fdfs_file_info /etc/fdfs/client.conf group1/M00/00/00/wKiJflaTKn-AfBqsAABdrZgsqUU861.jpg

5.删除

格式：fdfs_delete_file 配置文件 存储路径
sudo fdfs_delete_file /etc/fdfs/client.conf group1/M00/00/00/wKiJflaTKn-AfBqsAABdrZgsqUU861.jpg

四、安装nginx并配置fastdfs模块

1. 下载解压

```
1. # 如果链接失效则换地址下载解压
2. cd /usr/local/src
3. sudo wget
   http://sourceforge.net/projects/fastdfs/files/FastDFS%20Nginx%20Module%20Source%20Code/fastdfs-nginx-module_v1.16.tar.gz/download
4. sudo wget http://zlib.net/zlib-1.2.8.tar.gz
5. sudo wget http://iweb.dl.sourceforge.net/project/pcre/pcre/8.37/pcre-8.37.tar.gz
6. sudo wget http://nginx.org/download/nginx-1.9.7.tar.gz
7. sudo tar -zxvf download
8. sudo tar -zxvf nginx-1.9.7.tar.gz
9. sudo tar -zxvf pcre-8.37.tar.gz
10. sudo tar -zxvf zlib-1.2.8.tar.gz
```

2. 编译安装nginx+fastdfs模块

```
sudo ln -s /usr/include/fast* /usr/local/include/
cd /usr/local/src/nginx-1.9.7
sudo ./configure --prefix=/usr/local/nginx --add-module=/usr/local/src/fastdfs-nginx-module/src --with-pcre=/usr/local/src/pcr-8.37 --with-zlib=/usr/local/src/zlib-1.2.8
sudo make
sudo make install
```

3. 配置nginx-fastdfs模块

```
sudo cp /usr/local/src/fastdfs-nginx-module/src/mod_fastdfs.conf /etc/fdfs/
sudo vim /etc/fdfs/mod_fastdfs.conf
```

```
1. base_path=/tmp # 保存日志目录
2. tracker_server=192.168.137.210:22122 # tracker 服务器的 IP 地址以及端口号
3. storage_server_port=23000 # storage 服务器的端口号
```

```

4.  group_name=group1 # 当前服务器的 group 名
5.  url_have_group_name = true # 文件 url 中是否有 group 名
6.  store_path_count=1 # 存储路径个数, 需要和 store_path 个数匹配
7.  store_path0=/data/fastdfs/storage # 存储路径
8.  # 从文件扩展名查找文件类型 (nginx时为true, 记得加上这行)
9.  http_need_find_content_type=true
10. group_count = 1 # 设置组的个数
11. ...
12. # 然后在末尾添加分组信息, 目前只有一个分组, 就只写一个
13. [group1]
14. group_name=group1
15. storage_server_port=23000
16. store_path_count=1
17. store_path0=/data/fastdfs/storage

```

4.建立 M00 至存储目录的符号连接

```

1.  [root@csr ~]# sudo ln -s /data/fastdfs/storage/data /data/fastdfs/storage/data/M00
2.  [root@csr ~]# ll /data/fastdfs/storage/data/M00
3.  rw-rw-rw- 1 root root 26 10月 29 11:22 /data/fastdfs/storage/data/M00 -> /data/fastdfs/storage/data

```

5.FastDFS文件下载恢复原始文件名

`sudo vim /usr/local/nginx/conf/nginx.conf` # 在最后加入一段server

```

1.  server {
2.      server_name  csr.com;
3.      listen 80;
4.      location /group1/M00 {
5.          root /data/fastdfs/storage/data;
6.          if ($arg_attname ~ .*) {
7.              add_header Content-Disposition
"attachment;filename=$arg_attname";
8.          }
9.          ngx_fastdfs_module;
10.         # add_header "Content-Type" "application/octet-stream";
11.     }
12.     location / {
13.         deny all;

```

```
14.     }  
15. }
```

然后 `nginx -t` 无语法错误后重启

访问的时候加上 `?attname=filename` 即可下载，例如：

"本机测试"

五、分布式实践（多group多storage + Nginx前后端）

1.环境 + 各部件作用

由于硬件的限制，这里只用了3台虚拟机做演示，线上的架构应该把各组件分开；

Ubuntu 16.04: 3台 (虚拟机)

- 192.168.137.210: Nginx前端+后端(mod_fastdfs)，tracker+storage (group2组)
- 192.168.137.220: Nginx后端(mod_fastdfs)，tracker+storage (group1组)
- 192.168.137.221: Nginx后端(mod_fastdfs)，storage (group1组)

Nginx前端的作用：用于反向代理

- 如果是做图片站，可以在这里做下缓存 (proxy_cache)
- 如果是用于附件，则不需要考虑这些
- 如果是用于网盘，最好是不要做缓存，因为有敏感资源会很麻烦
- (或者可以把缓存时间设短，如1h；又或者编译Nginx的时候加上第三方模块 ngx_cache_purge可以手动清理缓存)

Nginx后端(作者建议每台机器都要安装一个)

- 编译时要加上 fastdfs-nginx-module 模块

多个tracker:

- 避免单点故障
- 配置上传到组的方式为：0-轮询

group1组(两个storage):

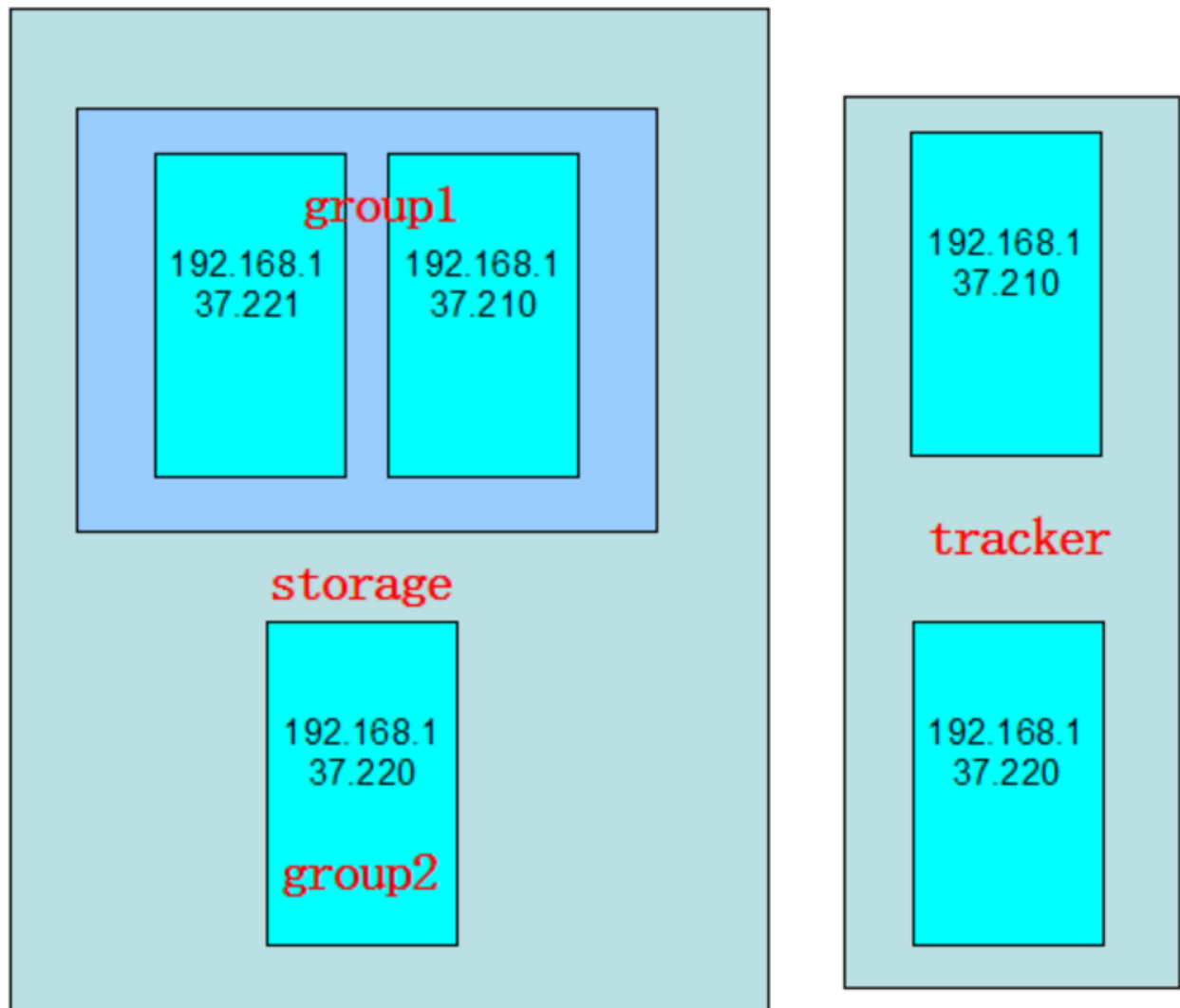
- 同组内的storage数据是相同的，互相备份
- 在storage目录下 data/.data_init_flag 有个 sync_src_server= 表示同步的server

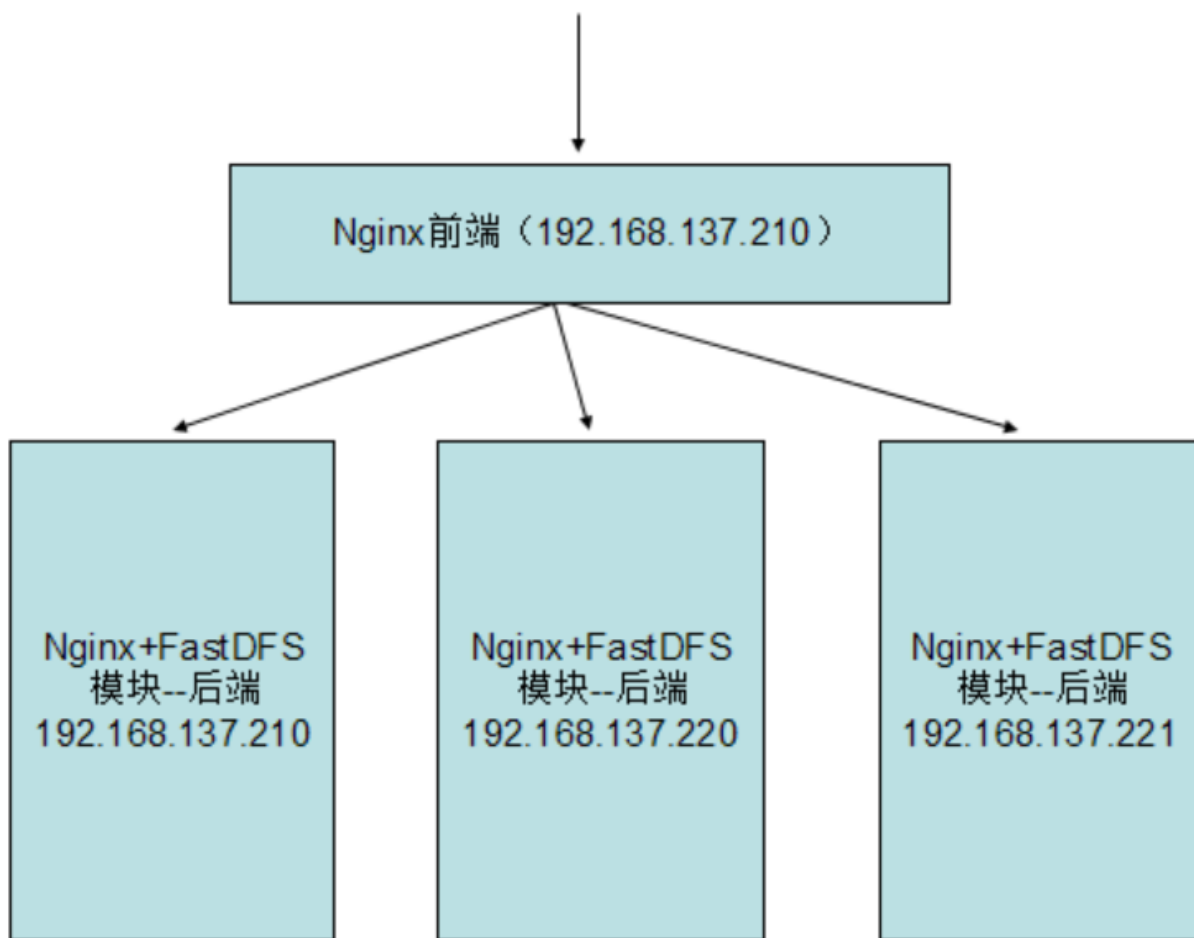
- 如果那个server down了，fdfs_monitor显示出来的状态会是 WAIT_SYNC
- 解决方法：先停止storage，然后删除这个文件，再启动storage即可

group2组:

- 多组，扩充容量

附上两张图片:





1. tracker + storage + Nginx后端安装同上，只是配置文件不同，下面直接贴配置
2. Nginx前端：随便编译安装个Nginx即可，然后加个代理配置
- 3.
4. 修改下面列出的配置，其他保持默认即可 (FastDFS默认配置文件位置：/etc/fdfs)

2.安装 + 配置 (192.168.137.210)

1.storage.conf

1. disabled=false # 启用配置文件
2. group_name=group1 # 组名
3. port=23000 # 设置 storage 的端口号
4. base_path=/data/fastdfs/storage # 设置 storage 的日志目录（需预先创建）
5. store_path_count=1 # 存储路径个数，需要和 store_path 个数匹配
6. store_path0=/data/fastdfs/storage # 存储路径
7. # tracker_server: 每个一行，可配多个 (不能设置成127.0.0.1)
8. tracker_server=192.168.137.210:22122

```
9. tracker_server=192.168.137.220:22122
10. http.server_port=8888 # 设置storage上启动的http服务的端口号 (即这里的Nginx后端端口号)
```

2.tracker.conf

```
1. disabled=false # 启用配置文件
2. port=22122 # 设置 tracker 端口号
3. base_path=/data/fastdfs/tracker # 设置 tracker 的数据文件和日志目录
4. store_lookup=0 # 指定上传到组的方式为 轮询方式
```

3.client.conf

```
1. base_path=/tmp
2. tracker_server=192.168.137.210:22122
3. tracker_server=192.168.137.220:22122
4. http.tracker_server_port=8888
```

4.mod_fastdfs.conf

```
1. base_path=/tmp # 保存日志目录
2. tracker_server=192.168.137.210:22122
3. tracker_server=192.168.137.220:22122
4. storage_server_port=23000 # storage 服务器的端口号
5. group_name=group1 # 当前服务器的 group 名
6. url_have_group_name = true # 文件 url 中是否有 group 名
7. store_path_count=1 # 存储路径个数, 需要和 store_path 个数匹配
8. store_path0=/data/fastdfs/storage # 存储路径
9. # 从文件扩展名查找文件类型 (nginx时为true, 记得加上这行)
10. http.need_find_content_type=true
11. group_count = 1 # 设置组的个数
12. [group1]
13. group_name=group1
14. storage_server_port=23000
15. store_path_count=1
16. store_path0=/data/fastdfs/storage
```

5.Nginx后端

```
1. worker_processes 1;
```

```

2.
3.  events {
4.      worker_connections 1024;
5.  }
6.
7.  http {
8.      include      mime.types;
9.      default_type  application/octet-stream;
10.
11.      sendfile      on;
12.
13.      keepalive_timeout 65;
14.
15.      server {
16.          listen      8888;
17.          server_name  localhost;
18.
19.          location / {
20.              root      html;
21.              index      index.html index.htm;
22.          }
23.
24.          location ~ /group[1-2]/M00 {
25.              root /data/fastdfs/storage/data;
26.              #         if ($arg_attname ~ .*) {
27.              #             add_header Content-Disposition
28.              #                 "attachment;filename=$arg_attname";
29.              #         }
30.              ngx_fastdfs_module;
31.              # add_header "Content-Type" "application/octet-stream";
32.          }
33.
34.          error_page 500 502 503 504 /50x.html;
35.          location = /50x.html {
36.              root      html;
37.          }
38.      }

```

6.前端Nginx

```

1.  worker_processes 1;
2.
3.  events {
4.      worker_connections 1024;
5.  }

```

```
6.
7.
8. http {
9.     include      mime.types;
10.    default_type  application/octet-stream;
11.
12.    sendfile      on;
13.
14.    keepalive_timeout 65;
15.
16.    upstream group1 {
17.        server 192.168.137.210:8888;
18.        server 192.168.137.221:8888;
19.    }
20.
21.    upstream group2 {
22.        server 192.168.137.220:8888;
23.    }
24.
25.    server {
26.        listen      80;
27.        server_name localhost;
28.
29.        location / {
30.            deny all;
31.        }
32.
33.        location /group1/M00 {
34.            proxy_pass http://group1;
35.            proxy_set_header Host $host;
36.            proxy_set_header X-Forwarded-For $remote_addr;
37.        }
38.
39.        location /group2/M00 {
40.            proxy_pass http://group2;
41.            proxy_set_header Host $host;
42.            proxy_set_header X-Forwarded-For $remote_addr;
43.        }
44.
45.        error_page 500 502 503 504 /50x.html;
46.        location = /50x.html {
47.            root html;
48.        }
49.    }
50. }
```

3.安装 + 配置 (192.168.137.220)

tracker client nginx 配置直接复制210机器的

storage.conf: 和210机器不一样的就只有group_name=group2
- group_name=group2 # 组是group2

mod_fastdfs.conf: (实际上和210不一样的也只有两行group_name)

```
1.  base_path=/tmp # 保存日志目录
2.  tracker_server=192.168.137.210:22122
3.  tracker_server=192.168.137.220:22122
4.  storage_server_port=23000 # storage 服务器的端口号
5.  group_name=group2 # 当前服务器的 group 名
6.  url_have_group_name = true # 文件 url 中是否有 group 名
7.  store_path_count=1 # 存储路径个数, 需要和 store_path 个数匹配
8.  store_path0=/data/fastdfs/storage # 存储路径
9.  # 从文件扩展名查找文件类型 (nginx时为true, 记得加上这行)
10. http_need_find_content_type=true
11. group_count = 1 # 设置组的个数
12. [group1]
13. group_name=group2
14. storage_server_port=23000
15. store_path_count=1
16. store_path0=/data/fastdfs/storage
```

4.安装 + 配置 (192.168.137.221)

storage client mod_fastdfs nginx后端 全部复制210机器配置

5.测试

首先fdfs_upload_file上传文件, 然后得到文件信息(如:

group1/M00/00/00/wKiJ0le5UVyAQwrrAAApWro0A0192.png)

- (因为是轮询的, 所以再次上传则会到group2组);

然后浏览器访

问: <http://192.168.137.210/group1/M00/00/00/wKiJ0le5UVyAQwrrAAApWro0A0192.png>

- 由于group1组有两个，并做了upstream负载均衡，所以会将请求负载均衡到210 221机器
 - 然后由后端Nginx处理请求并返回...
- (同样的，如果是group2组，前端则会发送至220机器的Nginx处理)

如果想在下载的时候恢复文件名，参照上面的内容配置即可

6.线上架构

线上的架构可以弄成这样：

两台Nginx前端机器 - HA

两台tracker，不需要做HA

N个group组(容量不够可扩容)，每个组2-3个storage(互备)

- 每个storage一台机器，同时每台都要搭建Nginx(mod_fastdfs)后端

配置和上面虚拟机实验的类似，稍微改些内容即可

参数需要优化之类的，可以到 CU社区 查查文档

根据用途的不同，做的优化也会有所不同，具体还是要结合实际情况的